Comparative studies on proximal algorithms for 3D MRI reconstruction

KAIST

2013
Comparative studies on proximal algorithms for 3D MRI reconstruction
Comparative studies on proximal algorithms for 3D MRI reconstruction

Advisor : Professor Ye, Jong Chul

by

Nam, Jae Hyun

Department of Bio and Brain Engineering

KAIST

A thesis submitted to the faculty of KAIST in partial fulfillment of the requirements for the degree of Master of Science in Engineering in the Department of Bio and Brain Engineering. The study was conducted in accordance with Code of Research Ethics.

2013. 7. 3.

Approved by

Professor Ye, Jong Chul

[Advisor]

---

1Declaration of Ethical Conduct in Research: I, as a graduate student of KAIST, hereby declare that I have not committed any acts that may damage the credibility of my research. These include, but are not limited to: falsification, thesis written by someone else, distortion of research findings or plagiarism. I affirm that my thesis contains honest conclusions based on my own careful research under the guidance of my thesis advisor.
3차원 MRI 영상 복원을 위한 proxy 근접 알고리즘 비교 연구

남 재 현

위 논문은 한국과학기술원 석사학위논문으로 학위논문심사위원회에서 심사 통과하였음.

2013년 06월 11일

심사위원장  예 중 철 (인)
심사위원  박 성 홍 (인)
심사위원  조 승 룡 (인)
ABSTRACT

Compressive sensing (CS) theory has been extensively investigated to reduce MRI scan time and reconstruct high resolution images from down-sampled k-space data by exploiting the sparsity of image in finite difference or transform domain. Recently, many high performance proximal algorithms such as a fast iterative shrinkage-thresholding algorithm (FISTA), alternating direction method of multipliers (ADMM), or a primal-dual algorithm have been considerably studied. However, depending on computing platforms, not only the rate of convergence but also memory usage are very important optimization factors. Hence, the main goal of this study is to perform comparative studies of these algorithms in terms of convergence speed and memory usages in the case of various type penalized 3D CS-MRI experiments including total variation, wavelet transform and patch-based low rank approximation. Our analysis and experimental results showed strength and weakness of each algorithm according to applications.
# Contents

Abstract .................................................. i
Contents .................................................. ii
List of Tables .......................................... iv
List of Figures .......................................... v

Chapter 1. Introduction .................................. 1

Chapter 2. Theory ....................................... 4
  2.1 Problem Formulation ................................. 4
  2.2 Proximal Algorithm ................................. 5
  2.3 Fast Iterative Shrinkage/Thresholding Algorithm .... 6
  2.4 Primal-dual Algorithm .............................. 7
  2.5 Alternating Direction Method of Multipliers ....... 9

Chapter 3. Implementation ............................. 11
  3.1 Implementation Information ...................... 11
  3.2 Implementation Process ............................. 12

Chapter 4. Total Variation Minimization .............. 14
  4.1 Algorithm description ............................. 14
  4.2 Demonstration ..................................... 17

Chapter 5. Wavelet Transform ......................... 23
  5.1 Algorithm Description ............................. 23
  5.2 Demonstration ..................................... 24

Chapter 6. Patch-based Low Rank Approximation .... 30
  6.1 Algorithm Description ............................. 30
  6.2 Demonstration ..................................... 33

Chapter 7. Discussion ................................... 36

Chapter 8. Conclusion .................................. 38

References .............................................. 39
List of Tables

4.1 Performance comparison of a primal-dual algorithm, FISTA and ADMM for reconstructing a down-sampled 3D Shepp-Logan image .................................................. 19
4.2 Performance comparison of a primal-dual algorithm, FISTA and ADMM for reconstructing a down-sampled MRA image .................................................. 22
5.1 Performance comparison of a primal-dual algorithm, FISTA and ADMM for reconstructing a down-sampled 3D Shepp-Logan image .................................................. 25
5.2 Performance comparison of a primal-dual algorithm, FISTA and ADMM for reconstructing a MRA image .................................................. 26
List of Figures

3.1 The original image of (a) a 64-th frame of a 3D Shepp-Logan phantom and (b) a 150-th frame of brain TOF MRA from full-sampling data ................................................. 11
3.2 Sampling pattern of phase encoding along y- and z- axis for (a) a phantom data using Gaussian random (b) brain MRA using uniform random ............................ 12
3.3 Maximum-intensity projections of (a) transaxial, (b) sagittal and (c) coronal planes, and revised images using a zero-filling procedures on (d) transaxial, (e) sagittal and (f) coronal 13
4.1 (a) 64-th frame of a down-sampled 3D Shepp-Logan phantom image by a factor of 3, and a reconstructed image using (b) a primal-dual algorithm, (c) FISTA and (d) ADMM ........................................ 18
4.2 mse curve versus computational time for a primal-dual algorithm (red), FISTA (blue) and ADMM (green) while reconstructing a down-sampled 3D Shepp-Logan image ............. 19
4.3 (a) 64-th frame of a down-sampled MRA image by a factor of 3, and a reconstructed image using (b) a primal-dual algorithm, (c) FISTA and (d) ADMM ........................................ 20
4.4 Maximum intensity projection images of (a) full-sampling data, (b) down-sampled data and reconstruction using (c) a primal-dual algorithm, (d) FISTA and (e) ADMM ........................................ 21
4.5 mse curve versus computational time for a primal-dual algorithm (red), FISTA (blue) and ADMM (green) while reconstructing a MRA image ................................................. 22
5.1 (a) 64-th frame of a down-sampled 3D Shepp-Logan phantom image by a factor of 3, and a reconstructed image using (b) a primal-dual algorithm, (c) FISTA and (d) ADMM ........................................ 25
5.2 mse curve versus computational time for a primal-dual algorithm (red), FISTA (blue) and ADMM (green) while reconstructing a down-sampled 3D Shepp-Logan image ............. 26
5.3 (a) 64-th frame of a down-sampled MRA image by a factor of 3, and a reconstructed image using (b) a primal-dual algorithm, (c) FISTA and (d) ADMM ........................................ 27
5.4 Maximum intensity projection images of (a) full-sampling data, (b) down-sampled data and reconstruction using (c) a primal-dual algorithm, (d) FISTA and (e) ADMM ........................................ 28
5.5 mse curve versus computational time for a primal-dual algorithm (red), FISTA (blue) and ADMM (green) while reconstructing a MRA image ................................................. 29
6.1 The scheme of finding similar patches .......................................................... 30
6.2 Low rank approximation .............................................................................. 31
6.3 (a) 64-th frame of an original 3D Shepp-Logan phantom image by a factor of 3, (b) down-damped image and (c) a reconstructed image using a primal-dual algorithm ........................................ 34
6.4 (a) 150-th frame of an original MRA phantom image by a factor of 3, (b) down-damped image and (c) a reconstructed image using a primal-dual algorithm ........................................ 34
6.5 Maximum intensity projection images of (a) full-sampling data, (b) down-sampled data and (c) reconstruction using a primal-dual algorithm ........................................ 35
7.1 Comparison of algorithm performance .......................................................... 37
Chapter 1. Introduction

Magnetic resonance imaging (MRI) is widely used in many clinical applications. One of the main advantages of MRI is that the internal parts of the body are obtained non-invasively without radiation exposure unlike X-ray CT. Instead of emitted radiation, the magnetization of protons (conventionally signified hydrogen in MR field) is contributed to generate images in MR system. When a proton is exposed to an external magnetic field, it is aligned along the direction of the field and rotated at the particular frequency by Larmor equation. Radio frequency (RF) pulse is applied at the same frequency of proton precession. Magnetic resonance is occurred to only a proton and it causes a proton to produce magnetization by RF pulse. As a result, a proton plays an exclusive role to construct images of the inner body. Furthermore, since the magnetization of a proton is different in types of molecules which a proton belongs to, the inner body is well visualized by good image contrast from magnetization gap between different tissues.

Frequency and phase are important factors to have an effect on magnetization. Because signal intensity is determined by magnetization, spatial localization can be accomplished by controlling two factors. When magnetic field gradient along one axis, usually x-axis, is applied, protons have slightly different precession frequency. This frequency encoding provides spatial localization along x-axis at one time. Localization along other axes using phase difference, however, cannot be performed simultaneously. Data along y-axis are acquired line by line in one phase encoding step. If region of interests (ROI) contains 128×128 voxels, 128 phase encoding steps are required to obtain fully sampled data. The number of phase encoding step along z-axis is also necessary for the same number of voxel on the z-axis in the case of the 3D data scan. In other words, the data acquisition is basically one dimensional process even though data is three dimensional. Hence, MRI is a relatively slow imaging modality.

Many researchers have studied to overcome a shortcoming of MR scan time. The problem of short scan time by skipping some phase encoding steps is that it is not possible to perform perfect image reconstruction due to partially missing k-space. In conventional image processing, aliasing in an image occurs when the data is down-sampled below the Nyquist frequency. Parallel imaging is one of the most important techniques to eliminate the aliasing problem. The down-sampled data is acquired using multiple receiver coils in MR system for parallel imaging. An image can be reconstructed better by compensating for incomplete k-space or image through various data and coil sensitivity of multiple coils. The compensating algorithms are divided into two groups. One group including SMASH (SiMultaneous Acquisition of Spatial Harmonics) [1] and GRAPPA (GeneRalized Autocalibrating Partially Parallel Acquisition) [2] reconstructs the global image by filling in missing k-space before Fourier transform. The other group such as SENSE (SENSitivity Encoding) [3] reconstructs aliasing parts by unfolding the image after Fourier transform. Even though these techniques have an advantage to reduce scan time, they suffer from the lower signal-to-noise ratio than an image from fully sampled data.

In dynamic MRI, a significant approach was introduced to improve spatio-temporal resolution using a priori information. k-t BLAST (Broad-use Linear Acquisition Speed-up Technique) and k-t SENSE are the representative algorithms [4]. k-t BLAST/SENSE does not require exact knowledge of the spectral support which gives advantages over the previous algorithms. A priori information from training data is used to reconstruct images in k-t BLAST/SENSE. With this information, lattice sampling is applied
Compressive sensing (CS) theory [5] has been studied as a breakthrough method to accelerate the MR acquisition. More specifically, from down-sampled k-space data below Nyquist frequency, accurate MR image reconstruction is possible if the sampling trajectory is incoherent and the signal is sparse in some bases. Even though the concept of CS theory already has been applied to image reconstruction of other bio-imaging modalities for a long time, it is used in MR field recently after introducing k-t Sparse and k-t FOCUSS (FOCal Underdetermined System Solver) [6]. Especially, a solution using k-t FOCUSS is represented as exactly same mathematical formula of the solution using k-t BLAST/SENSE although it is not required training set unlike k-t BLAST/SENSE. FOCUSS was originally developed in EEG field to obtain a sparse solution to a quadratic optimization problem [7]. The algorithm applying FOCUSS in dynamic MR is called k-t FOCUSS because it finds a sparse solution in x-t domain from down-sampled measurements in k-t domain. It is empirically explained that an image is well-reconstructed using k-t BLAST/SENSE. However, k-t FOCUSS analytically explained the reason of its high performance by introducing CS theory. Even the first iteration of k-t FOCUSS is identical to k-t BALST/SENSE. After the introduction of CS theory with remarkable results of image reconstruction in dynamic MR, many researchers have tried to develop novel algorithms using CS theory in various MR fields.

A basic principle of CS theory is to find a unique sparse solution to an underdetermined system. Recent studies revealed that low rank approximation has a similar role with sparsity in CS theory. It is expected that a low rank signal are constituted of high coherent vectors. This characteristic of the low rank signal can be used for sparse representation of the signal. A barrier to apply CS theory is that it is difficult to solve an optimization problem including sparsity or low rank term because they are represented as convex, but non-differentiable functions. Smooth convex optimization problems can be easily solved by directly differentiating functions or using gradient methods. However, CS problems cannot be differentiated at a global minimum. Many algorithms are developed to overcome this difficulty such as iterative shrinkage/thresholding algorithm (ISTA) [8], gradient projection method and proximal minimization algorithms. The disadvantages of these algorithms are that they are applied to only specific cases and show slow convergence rate.

Faster and more efficient methods are required to deal with high dimensional large volume data. Many studies showed the extended algorithms to find a sparse or low rank solution in a general case with fast convergence rate. Three algorithms currently have attracted a lot of attention in an image processing community. In fast iterative shrinkage/thresholding algorithm (FISTA) [9], a two-step iterative algorithm dramatically improves the convergence rate of the ISTA even though a simple computational effort is conserved. However, the previous studies of FISTA did not show an algorithm for finding a solution in a general case. Alternating direction method of multipliers (ADMM) is the representative of augmented Lagrangian (AL) techniques [10]. In an AL algorithm, a vector of Lagrange multipliers is included as a penalty parameter for satisfying equality constraints in an optimization problem. ADMM is combined with variable splitting to apply AL to more general cases. The primal-dual optimization algorithm recently receives attention as a novel method using primal minimization and dual maximization to directly solve a saddle point problem [11]. The main advantage of a primal-dual algorithm is the wide applicability with fast convergence rate since a solution to the general problem can be derived using the algorithm.

In this study, we review these algorithms and discuss their relationship. While the inter-relationships between these algorithms have been studied in mathematical literatures, we are not aware of one in biomedical imaging literature. Therefore, main goals of this study are 1) to provide introductory de-
criptions of the differences and similarities of these algorithms, and 2) to analyze their pros and cons in the perspectives of the convergence level, speed and memory usage of FISTA, ADMM and a primal-dual algorithm for solving a dictionary operator regularized 3D MRI problem. Memory usages are also one of the important criteria since memory usages are often more critical factors due to their limited bandwidth. In addition, we derive a general application of FISTA by applying a primal-dual algorithm to FISTA. Chapter 2 explains FISTA, ADMM and a primal-dual algorithm in detail in a general framework with a dictionary operator. In Chapter 3, data information and implementation process is described. We adapt these three proximal algorithms to total variation, wavelet transform and patch-based low rank approximation. Algorithm description and demonstration results of image reconstruction using three applications are shown in Chapter 4, 5 and 6, respectively, and we discuss and analyze results in Chapter 7. Finally, Chapter 8 provides conclusion.
Chapter 2. Theory

2.1 Problem Formulation

A basic unconstrained optimization problem is formulated as following:

$$\min_x f_1(x) + f_2(x), \quad (2.1)$$

where $f_1(x)$ and $f_2(x)$ denote a fidelity term and a regularization term in CS-MRI problems, respectively. A fidelity term is usually considered as $f_1(x) = \|Ax - b\|^2_2$ where $b \in R^M$ is down-sampled k-space data, $A \in R^{M \times N}$ is the matrix representation of under-sampled Fourier transform, and $x \in R^N$ is unknown image in CS-MRI reconstruction problem.

The goal of a CS problem is to find sparse approximation of $x$, which means that $x$ contains many zero elements. A sparse solution can be found from $\|x\|_0$ norm minimization problem:

$$\min_x \|x\|_0 \quad \text{subject to} \quad b = Ax. \quad (2.2)$$

Eq.2.2 is NP-hard (Non-deterministic Polynomial-time hard) problem for large $n$ because $\|x\|_0$ is obtained by counting the number of nonzero elements in $x$. $l_1$ norm can play the analogous role of $l_0$ norm when the matrix $A$ satisfies the restricted isometry property (RIP) condition [5]. Using this property, eq.2.1 can be substituted for the following unconstrained problem:

$$\min_x \|Ax - b\|^2_2 + \lambda \|x\|_1, \quad (2.3)$$

where $\lambda$ is a regularization parameter.

Low rank approximation is an alternative to sparsity in a CS technique. The rank of a matrix is defined as the number of linearly independent row or column vectors. A sparse signal is composed of correlated vectors so that it has few numbers of independent vectors causing low rank. It is expected that the quality of a reconstructed signal is dramatically improved by using low rank approximation,

$$\min_X \text{rank}(X) \quad \text{subject to} \quad B = AX, \quad (2.4)$$

where $B$ and $X$ are matrices. Since eq.2.4 is also NP-hard problem like eq.2.2, we instead solve a transformed unconstrained optimization problem

$$\min_X \|AX - B\|^2_F + \lambda \|X\|_* \quad (2.5)$$

where $A$ matrix follows RIP condition, and $\|X\|_*$ is sum of singular values of $X$, nuclear norm. In other words, nuclear norm is expressed as the form of $l_1$ norm of singular values, $\|X\|_* = \sum_i \eta_i(X)$ where $\eta_i(X)$ is $i$-th singular value of $X$. Hence, eq.2.5 is easily solved in a similar way as solving eq.2.3 by using proximal mapping and soft-thresholding after applying singular value decomposition (SVD) to $X$ in eq.2.5.

It is easy to reconstruct an image by solving eq.2.3 and 2.5 when an image is originally sparse. However, since most of the inner parts of a body is not sparse, researchers have studied to transform a bio-medical image into the domain which shows sparse representation. For example, $x$ is transformed
into $x - f$ domain in dynamic MRI due to the similarity of images along time frames. In most of static states, $x$ does not have sparse representation in itself so that operators such as TV are introduced in order to employ CS technique. Then, $x$ in the regularization function of eq.2.1 is substituted for $Dx$

$$
\min_x f_1(x) + f_2(Dx),
$$

where $D$ is a dictionary operator matrix. Algorithms have been developed to find a sparse approximation of $x$ for the more general case of eq.2.6 in the corresponding transform domain, $D$.

### 2.2 Proximal Algorithm

The main barrier to solve CS-MRI problems is that $f_2$ in eq.2.6 is non-differentiable. Various algorithms, such as steepest decent method and conjugate gradient method, have been developed to find a solution of smooth optimization problems. These algorithms search a global minimum point by finding the critical point of a problem. However, it is difficult to find this point in non-differentiable functions; e.g. $l_1$ norm and nuclear norm. Therefore, another approach is required to solve non-smooth optimization problems.

To overcome non-differentiable difficulty, the concept of a proximal algorithm is introduced. In proximal algorithms, the resolvent operator is used as an alternative of finding a critical point. The resolvent of a function $F$, $R$, is defined as

$$
R = (I + \eta F)^{-1}
$$

When the subdifferential of $f$ contains 0, $\partial f(x) \ni 0$, the following relations are satisfied:

$$
\partial f(x) \ni 0 \iff x + \eta \partial f(x) \ni x \iff x = (I + \eta \partial f)^{-1}(x) \iff x = R(x).
$$

These relations show that it is possible to find a solution using a resolvent operator instead of a derivative. In addition, a resolvent operator is nonexpansive when $f$ is convex. It means that $R(x_n)$ converges to some point as $n$ goes to infinity. That is, a converging unique solution can be obtained using resolvent in convex optimization problems by $R(x_n) = x_{n+1}$.

Proximal algorithms are implemented using the proximity operator of a non-smooth function [12]. A proximity operator comes from the following relations:

$$
x_{n+1} = R(x_n) = (I + \eta \partial f)^{-1}(x_n)
$$

$$
\iff x_{n+1} + \eta \partial f(x_{n+1}) \ni x_n
$$

$$
\iff 0 \in \partial x_{n+1}(f(x_{n+1}) + \frac{1}{2\eta} \|x_{n+1} - x_n\|^2_2)
$$

$$
\iff x_{n+1} = \arg \min_x (f(x) + \frac{1}{2\eta} \|x - x_n\|^2_2) = \text{prox}_\eta[f](x_n),
$$

where $\eta$ is a step-size. It is comfortable to explicitly express the operator because it has a distinct form according to a type of the convex function. When we want to find a global minimum of $l_1$ norm, $f(x) = \|x\|_1$, using the proximity operator, the solution is obtained as the following form:

$$
x_{n+1} = \text{prox}_\eta[f](x_n) = \arg \min_x \|x\|_1 + \frac{1}{2\eta} \|x - x_n\|^2_2
$$

$$
= \begin{cases} 
(|x_n|_i - \eta) \times \text{sign}(|x_n|_i), & \text{if } |x_n|_i \geq \eta \\
0, & \text{otherwise}
\end{cases}
$$

Using the concept of a proximity operator, variants of proximal algorithms have been introduced to find a solution of non-smooth optimization problems with the fast convergence rate for many applications.
2.3 Fast Iterative Shrinkage/Thresholding Algorithm

A forward backward algorithm is one of the representative proximal algorithms. This algorithm is used for the case of two-term problems where one is smooth and the other is non-smooth like eq.2.6 [13]. \(f_1\) is a differentiable fidelity term and \(f_2\) is usually a non-differentiable penalty term in CS-MRI problems. In this case, a proximal algorithm searches a global minimum iteratively:

\[
x_{n+1} = \text{prox}_{\eta f_2}(x_n - \eta \nabla f_1(x_n))
\]

\[
= \arg \min_x \left\{ \frac{1}{2\eta} \|x - (x_n - \eta \nabla f_1(x_n))\|^2_2 + f_2(Dx) \right\}.
\]

(2.11)

Here, the step in a bracket is a forward gradient step while the other, a proximity operator, is a backward step. Hence, the algorithm is called a forward-backward algorithms.

ISTA is one of the forward-backward algorithm. In ISTA, the Lipschitz constant of \(\nabla f_1\) is introduced instead of \(\eta\) in the role of a step-size where for every \(x_1, x_2 \in R^N\). The Lipschitz constant is defined as

\[
L(f_1) \geq \frac{\|\nabla f_1(x_1) - \nabla f_1(x_2)\|_2}{\|x_1 - x_2\|_2}.
\]

(2.12)

ISTA searches the global minimum of eq.2.11 through the following iterative steps.

**Algorithm1. ISTA**

**Input :** \(L\): the Lipschitz constant of \(f_1\).

**Step 0.** Initialize \(x_1\).

**Step** \(n\).

\[
\tilde{x}_n = x_n - \frac{1}{L} \nabla f(x_n),
\]

(2.13)

\[
x_{n+1} = \text{prox}_{\frac{1}{L} f_2}(\tilde{x}_n).
\]

(2.14)

Each step in ISTA is a quite simple and low cost algorithm because proximal mapping of eq.2.14 is free from \(A\) matrix after performing eq.2.13. In low computational effort, the algorithm guarantees that \(x_{n+1}\) always goes to the global minimum by the inequality

\[
C(x_{n+1}) - C(x^*) \leq \frac{L(f_1)}{2\eta} \|x_1 - x^*\|^2_2
\]

for any \(n \geq 1\),

(2.15)

where \(C\) denotes the whole cost function in eq.2.6. Eq.2.15 implies that \(F(x_{n+1})\) converges to the minimum value \(C(x^*)\) at a rate of convergence \(O(1/k)\).

FISTA is an improved version of ISTA with super-linear convergence rate while keeping ISTA’s simplicity [9]. FISTA iterates the following steps.

**Algorithm2. FISTA**

**Input :** \(L\): the Lipschitz constant of \(f_1\).

**Step 0.** Initialize \(x_1, y_1 = x_1, t_1 = 1\).

**Step** \(n\).

\[
\tilde{y}_n = y_n - \frac{1}{L} \nabla f(y_n),
\]

(2.16)

\[
x_{n+1} = \text{prox}_{\frac{1}{L} f_2}(\tilde{y}_n),
\]

(2.17)

\[
t_{n+1} = \frac{1 + \sqrt{1 + 4t^2}}{2}
\]

(2.18)

\[
y_{n+1} = x_{n+1} + \left(\frac{t_n - 1}{t_{n+1}}\right) (x_{n+1} - x_n).
\]

(2.19)

FISTA is different from ISTA introducing a new variable \(y_n\) for the proximal mapping while ISTA uses the variable \(x_n\). In ISTA, \(x_n\) is used in the proximal mapping which is obtained in the \((n - 1)\)-th
step. In FISTA, however, the point $y$ obtained by the linear combination of $x_{n+1}$ and $x_n$ of eq.2.19 is used for the convergence. The linear combination searches a point nearer to the optimal point $x^*$ in each step. Hence, the complexity rate is accelerated to $O(1/k^2)$ compared to ISTA by the inequality

$$C(x_{n+1}) - C(x^*) \leq \frac{2L(f_1)}{(n+1)^2} \|x_1 - x^*\|^2$$

for any $n \geq 1$. (2.20)

For each extra-gradient step of $x$, the weighting factor is calculated accurately. However, there is a trade-off between the convergence rate and the memory required compared to ISTA. FISTA needs to store the previous two points and the point from a linear combination for an extra-gradient step. Hence, the algorithm demands memory space to store $x_n$, $x_{n+1}$ and $y$. The linear combination in FISTA requires only a few computational effort so that it is negligible in the view of time complexity. The main advantage of FISTA is dramatically faster convergence rate than previous algorithms while keeping simplicity. But, there is a limitation for applying the algorithm to more general cases.

### 2.4 Primal-dual Algorithm

The primal-dual algorithm solves an optimization problem by dealing with a primal minimization and a dual maximization problem simultaneously. When a primal problem is considered as an original optimization problem of eq.2.6, a dual problem is defined as

$$\max_q -f_1^*(D^Hq) - f_2^*(q)$$

where $q$ is linearly transformed vector from $x$ in the space $X$ into the space $Q$ by the matrix $D$ and $^*$ means convex conjugate of a function determined by Legendre-Fenchel transformation [14] of

$$F^*(q) = \max_x \{q^T(Dx) - F(Dx)\},$$

$$F(Dx) = \max_q \{q^T(D^Hq) - f^*_2(q)\}.$$  (2.22) (2.23)

Introducing an additional dual problem eq.2.21 seems to be a burden. However, the dual plays a highly important role in a primal-dual algorithm. As the difference between the primal and dual equations goes to zero, the algorithm guarantees the convergence of function value at a rate of $O(1/n^2)$. Furthermore, it can be applied to the general case of an optimization problem by converting a non-smooth function, which is hard to solve, to a solvable conjugate function of a dual problem.

A primal and dual problem can be merged into the following saddle-point problem

$$\min_x \max_q f_1(x) + \langle q, Dx \rangle - f_2^*(q).$$

(2.24)

The distance between the cost function value of a primal and dual problem, the duality gap, becomes much closer by repeatedly solving both problems. The proximal mapping in eq.2.9 can be used for two optimization problems. Then, the variables $q$ and $x$ are iteratively updated by implementing proximal mapping on $f_1$ and $f_2^*$

$$q_{n+1} = prox_{\sigma}[-C_{\text{saddle}}(q_n)],$$

$$= \arg \min_q \left\{ \frac{1}{2\sigma} \|q - q_n\|^2 - f_1(x_n) - \langle q, Dx_n \rangle + f_2^*(q) \right\},$$

$$= \arg \min_q \left\{ \frac{1}{2\sigma} \|q - q_n - \sigma Dx_n\|^2 + f_2^*(q) \right\},$$

$$= prox_{\sigma}[f_2^*](q_n + \sigma Dx_n),$$

(2.25)
With these conditions, a primal-dual algorithm is introduced below.

The algorithm iterates infinitely. This can be shown by the following inequality,

\[
\begin{align*}
\|q_n - q_N\|_2^2 &+ \frac{\|x_n - x_N\|_2^2}{2\tau} + (1 - \sqrt{\sigma \tau K}) \sum_{k=n_k}^{N-1} \frac{\|q_k - q_{k-1}\|_2^2}{2\tau} \leq \frac{\|x_n - x_{n_k}\|_2^2}{2\tau} + \frac{\|x_N - x_{N-1}\|_2^2}{2\tau} + (D(x_N - x_N - 1), q_N - q^*) \\
&+ (1 - \sqrt{\sigma \tau K}) \sum_{n=n_k}^{N-1} \frac{\|x_n - x_{n-1}\|_2^2}{2\tau} \leq \frac{\|x_n - x_{n_k}\|_2^2}{2\tau} + \frac{\|x^* - x_{n_k}\|_2^2}{2\tau}.
\end{align*}
\]

(2.30)

Where \(\|\cdot\|_2\) is the objective function in eq.2.24 and \((q, Dx) = (x, Dhq)\). \(\sigma\) and \(\tau\) are step-sizes for eq.2.25 and 2.26 respectively, like the role of \(L\) in eq.2.17. They are determined by \(K = \|D\|_2\) according to the theorem which explains that \((x_n, q_n)\) goes to a saddle-point \((x^*, q^*)\) when \(\sigma \tau K^2 < 1\) and \(\theta = 1\). With these conditions, a primal-dual algorithm is introduced below.

**Algorithm 3. Primal-Dual Algorithm**

**Input:** \(K = \|D\|_2\), \(\sigma = \frac{1}{K}, \tau = \frac{1}{K}, \theta = 1\).

**Step 0.** Initialize \(x_1\) and \(q_1\), \(x_1 = x_1\).

**Step n.**

\[
\begin{align*}
q_{n+1} &= \text{prox}_\sigma[f_2](q_n + \sigma Dx_n), (2.27) \\
x_{n+1} &= \text{prox}_\tau[f_1](x_n - \tau Dhq_{n+1}), (2.28) \\
x_{n+1} &= x_{n+1} + \theta(x_{n+1} - x_n). (2.29)
\end{align*}
\]

Eq.2.29 and \(\theta\) play similar role to the linear combination of eq.2.19 in FISTA. The linear combination also performs the additional gradient step so that superlinear order convergence can be also achieved using a primal-dual algorithm by choosing \(\theta\) similar to FISTA. However, \(\theta\) is fixed in a primal-dual algorithm while \(t\) is updated by eq.2.19 in every step. Note that a dual variable \(q\) usually works as a weighting factor for the regularization term, so to achieve fast convergence we need a sufficiently accurate primal variable to calculate a meaningful weighting factor for the regularization term. From a memory view, a primal-dual algorithm uses space to keep \(x_n, x_{n+1}, x_{n+1}\) and additionally, \(q_{n+1}\). The primal-dual algorithm requires more memory space not only for an extra-gradient step but also for a dual variable. That is, compared to FISTA, the price of memory usage should be paid for the extended application by introducing \(D\).

In [11], it is proved that \(x_n\) and \(q_n\) in a primal-dual algorithm goes to a saddle point, \((x^*, q^*)\) as the algorithm iterates infinitely. This can be shown by the following inequality,

\[
\frac{\|q - q_N\|_2^2}{2\sigma} + \frac{\|x - x_N\|_2^2}{2\tau} + (1 - \sqrt{\sigma \tau K}) \sum_{n=n_k}^{N-1} \frac{\|q_n - q_{n-1}\|_2^2}{2\tau} \leq \frac{\|x_n - x_{n_k}\|_2^2}{2\tau} + \frac{\|x_N - x_{N-1}\|_2^2}{2\tau} + (D(x_N - x_N - 1), q_N - q^*) \\
+ (1 - \sqrt{\sigma \tau K}) \sum_{n=n_k}^{N-1} \frac{\|x_n - x_{n-1}\|_2^2}{2\tau} \leq \frac{\|x_n - x_{n_k}\|_2^2}{2\tau} + \frac{\|x^* - x_{n_k}\|_2^2}{2\tau}.
\]

(2.30)
2.5 Alternating Direction Method of Multipliers

ADMM belongs to a family of AL methods combined with variable splitting. AL is another type of a proximal algorithms which tries to solve constrained optimization problems [15]. In the view of image processing, eq.2.1 is reformulated as a constrained problem of

\[
\min_x f_2(x) \quad \text{subject to} \quad b = Ax. \tag{2.31}
\]

A solution is a point where the curve of \( f_2 \) tangentially touches the constraint curve of \( Ax = b \). It means that the gradients of \( f_2 \) and \( Ax \) are parallel on the solution point. Thus, we formulate an unconstrained auxiliary function:

\[
L(x, w) = f_2(x) + w^H(Ax - b), \tag{2.32}
\]

where \( w \), a vector of Lagrange multipliers, is required to complement the unequal magnitudes of two parallel gradient vectors. Then, a solution of eq.2.31 becomes a critical point of \( L \) and \( \nabla_w L(x, w) = 0 \) means \( Ax = b \).

It is similar way with a proximal approach to find a critical point of \( L \). Because the constrained condition and \( \nabla_x L \ni 0 \) should be satisfied, we have

\[
0 \in \partial f_2(x_n+1) + A w_n^H = \partial f_2(x_n+1) + A^H w_n + \rho A^H (Ax_n+1 - b) = \partial f_2(x_n+1) + A^H (w_n + \rho(Ax_n+1 - b)) = \partial f_2(x_n+1) + A^H w_{n+1}, \tag{2.33}
\]

\[
0 \in \partial f_2(x_n+1) + A^H w_{n+1} = \partial f_2(x_n+1) + \rho(Ax_n+1 - b) \tag{2.34}
\]

Eq.2.33 and 2.34 are used for updating a primal variable \( x \) and a dual variable \( w \) respectively:

<table>
<thead>
<tr>
<th>Algorithm 4. Augmented Lagrangian Method</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 0.</strong> Initialize ( x_1, w_1 ).</td>
</tr>
<tr>
<td><strong>Step n.</strong></td>
</tr>
<tr>
<td>( x_{n+1} = \arg \min_x f_2(x) + w_n^H (Ax - b) + \frac{\rho}{2} |Ax - b|_2^2, \tag{2.35} )</td>
</tr>
<tr>
<td>( w_{n+1} = w_n + \rho(Ax_{n+1} - b) \tag{2.36} )</td>
</tr>
</tbody>
</table>

A vector of Lagrangian multipliers improves the accuracy of estimation of a solution at each iteration so that AL method brings out rapid convergence rate. Even though derivation of the rate of convergence in a general case has not been proved, some studies showed analytically the rate of convergence in specific cases and the method is usually implied to exponential method of multipliers. The convergence rate in a general case is estimated to show the super-linear order convergence. A vector of Lagrangian multipliers results in another main advantage of the method which is that \( \rho \) is not required to take infinity for convergence unlike the penalty methods. In addition, the initialization of \( x \) for convergence is less critical so that a solution converges under more general cases.

AL methods have a weak point in a general non-smooth problem including ill-conditioned matrices like eq.2.6 even though it has many advantages such as fast rate of convergence and less sensitiveness in initialization. To overcome this weakness, variable splitting is introduced to AL methods. The unconstrained problem of eq.2.6 is modified to the following new constrained problem by substituting \( Dx \), a main trouble, to a new variable \( z \) [16],

\[
\min_{x,z} f_1(x) + f_2(z) \quad \text{subject to} \quad z = Dx. \tag{2.37}
\]
The constrained problem of eq.2.37 can be solved by AL methods as the similar way by converting eq.2.31 to eq.2.32,

\[
\mathcal{L}(x, z, w, \rho) = f_1(x) + f_2(z) + w^H(z - Dx) + \frac{\rho}{2}\|Dx - z\|^2. \tag{2.38}
\]

The cost function of a problem is separable into \(x\) and \(z\) so that dual variables are updated independently by solving for one variable while the other is fixed. Then, a vector of Lagrangian multipliers is updated.

\begin{algorithm}
\textbf{Algorithm 5. Alternating Direction Method of Multipliers}
\begin{align*}
\text{Step 0.} & \quad \text{Initialize } x_1, z_1 = Dx_1, w_1. \\
\text{Step } n. & \quad x_{n+1} = \arg\min_x f_1(x) + \frac{\rho}{2}\|Dx - z_n - w_n'\|^2, \tag{2.39} \\
& \quad z_{n+1} = \arg\min_z f_2(z) + \frac{\rho}{2}\|Dx_{n+1} - z - w_n'\|^2, \tag{2.40} \\
& \quad w_{n+1}' = w_n' - (Dx_{n+1} - z_{n+1}). \tag{2.41}
\end{align*}
\end{algorithm}

In the algorithm scheme, \(w'\) is a scaled vector of Lagrangian multipliers of \(w\) to insert an augmentation term into a quadratic term. The low rank of ADMM for convergece is that \(D\) should have full column rank. Then, convergence of ADMM is shown by Eckstein and Bertsekas [15]. When \(D\) has full column rank, and if \(f_1\) and \(f_2\) are closed, proper, and convex functions, it is satisfied that

\[
\|x_{n+1} - \arg\min_x f_1(x) + \frac{\rho}{2}\|Dx - z_n - w_n'\|^2 \| \leq \alpha_n \tag{2.42}
\]

\[
\|z_{n+1} - \arg\min_z f_2(z) + \frac{\rho}{2}\|Dx_{n+1} - z - w_n'\|^2 \| \leq \beta_n \tag{2.43}
\]

\[
w_{n+1}' = w_n' - (Dx_{n+1} - z_{n+1}), \tag{2.44}
\]

where \(\alpha_n\) and \(\beta_n\) are sequences such that \(\sum_{n=1}^{\infty} \alpha_n < \infty\) and \(\sum_{n=1}^{\infty} \beta_n < \infty\). Then, \(x_n\) eventually converges to a solution if eq.2.37 has a solution.

ADMM can be used in more applications than basic AL scheme while it preserves lots of advantages of AL methods. In memory storage perspective, ADMM pays the extra memory price for a vector of Lagrangian multipliers and variable splitting, \(w'\) and \(z\) respectively. This memory usage brings out the merit of ADMM. The algorithm requires less memory space than FISTA and a primal-dual although \(x_n\) converges to a solution with the super-linear order because it does not take a linear combination unlike other two algorithms. ADMM is one of the widely studied optimization algorithms by reason of these characteristics.
Chapter 3. Implementation

3.1 Implementation Information

All algorithms are first tested if they works well or not with a $128 \times 128 \times 128$ 3D Shepp-Logan phantom. After the phantom demonstration, we applied all algorithms with various regularization terms to time-of-flight (TOF) MR angiogram (MRA) of the brain acquired by the conventional single-echo. The single-echo MRA data were scanned on a 3-T whole-body scanner (Siemens Medical Solutions, Erlangen, Germany) with a body-coil transmission and a 12-element head matrix coil reception. Voxel-localized shimming was performed with a vendor-supplied shim module based on a three-dimensional (3D) phase map. The datasets were acquired in a single slab. Imaging parameters were as follows: pulse repetition time = 40 ms, acquisition bandwidth = 150/50Hz/pixel, matrix size = $512 \times 192 \times 64$, corresponding field of view = $240 \times 180 \times 80$ mm$^3$, and number of averages = 1. 20% of a slice oversampling was employed to avoid a wraparound artifact. The scan time for a 3D dataset was 9.8 min. All the experiments were implemented using MATLAB on a computer equipped with an Intel i3-2100 3.10GHz processor, with 4GB of RAM. Fig.3.1 shows a frame of original images of a 3D Shepp-Logan phantom and brain TOF MRA from full-sampling data.

Figure 3.1: The original image of (a) a 64-th frame of a 3D Shepp-Logan phantom and (b) a 150-th frame of brain TOF MRA from full-sampling data.
3.2 Implementation Process

The main goal of 3D CS-MRI is to reduce a scan-time while acquiring high quality of an image compared to the conventional method. For this aim, we implemented artificially down-sampled data by generating a sampling pattern mask in fig.3.2. Since frequency encoding obtains information of all voxels along the axis at once, there is no time loss during frequency encoding. Most of scan-time cost results from phase-encoding. That is, the number of voxels where the value is lost means that the reduced scan-time during phase-encoding. Hence, we considered that spatial localization along each axis of masks in fig.3.2 is obtained using phase-encoding steps and the mask is multiplied along x-axis because of full-sampling by the frequency encoding step. In \( k \)-space, signals in low frequencies keep important information to formulate an image including signal to noise and contrast information while the signal in the region of high frequencies contains less important information affecting the resolution of the image. It is attempted to obtain information near the center of \( k \)-space so that Gaussian random sampling is adopted like fig.3.2(a). However, the purpose of MRA is to show blood vessels whose information is in high frequencies. For this reason, fig.3.2(b) is produced using uniform random sampling.

![Sampling pattern of phase encoding along y- and z-axis for (a) a phantom data using Gaussian random (b) brain MRA using uniform random](image)

Figure 3.2: Sampling pattern of phase encoding along y- and z- axis for (a) a phantom data using Gaussian random (b) brain MRA using uniform random

The in vivo data in this study is obtained by 4 different coils. Because the coil sensitivity is unknown, coil images are combined by using sum of squares. MRA brain images were displayed maximum-intensity projections (MIPs) to focus on blood vessels. Fig.3.3 (a), (b) and (c) show MIPs of the transaxial, sagittal and coronal planes, respectively. MIPs are displayed discretely since the rate of axes of field of view is different with that of an image size. Consequently, a zero-filling procedure is required for the isotropic voxel dimension and better image quality like fig.3.3 (d), (e) and (f). A 3D raw dataset from each RF channel was Fourier-transformed to generate 3D images with the matrix size of 512 \( \times \) 400 \( \times \) 206.

Mean square error (MSE) of an intermediate reconstruction, \( x_n \) is defined by

\[
MSE_n = \frac{\|x_n - x_{true}\|^2}{\|x_{true}\|^2},
\]

where \( x_n \) is a reconstructed image in \( n \)-th iteration and \( x_{true} \) is an original image from full-sampling data. Eq.3.1 is used for calculating the convergence rate.
Figure 3.3: Maximum-intensity projections of (a) transaxial, (b) sagittal and (c) coronal planes, and revised images using a zero-filling procedures on (d) transaxial, (e) sagittal and (f) coronal
Chapter 4. Total Variation Minimization

4.1 Algorithm description

Recently, optimization problems with total variation (TV) have received a lot of attention for MR image reconstruction because of their potential to provide high quality images from sparse view sampling. TV has been known to be useful for performing edge-preserving regularization, and recent developments in CS have sparked even greater interest [17], [18], [19]. However, a problem with TV is difficult to solve because of its non-smoothness even though the term is convex. This difficulty can be overcome by algorithms using $l_1$ norm because TV is a type of $l_1$ norm. The definition of TV is sum of the difference between neighbor voxels. When it comes to a problem for reconstruction of a 3D image, $x \in \mathbb{R}^{I \times J \times K}$, a gradient operator $\nabla$ is defined as:

$$\nabla x = \begin{pmatrix} \nabla_I x \\ \nabla_J x \\ \nabla_K x \end{pmatrix},$$

(4.1)

where

$$\nabla_I x = \begin{cases} x_{i+1,j,k} - x_{i,j,k} & i < I \\ -x_{i,j,k} & i = I \end{cases}, \quad \nabla_J x = \begin{cases} x_{i,j+1,k} - x_{i,j,k} & j < J \\ -x_{i,j,k} & j = J \end{cases}, \quad \nabla_K x = \begin{cases} x_{i,j,k+1} - x_{i,j,k} & k < K \\ -x_{i,j,k} & k = K \end{cases}.$$  

(4.2)

Consequently, $\text{TV}(x) = \|\nabla x\|_1$ and eq.2.6 is reformulated as following:

$$\min_x \|Ax - b\|^2_2 + \lambda \|\nabla x\|_1$$

(4.3)

where $\nabla$ becomes a transformation operator $\nabla$. The dimension of $\nabla x$ is three times as big as the size of $x$. In this case, $\nabla^H$ is obtained as the following form:

$$-\text{div}(\nabla x) = -(\nabla_I x_{i,j,k} - \nabla_I x_{i-1,j,k}) - (\nabla_J x_{i,j,k} - \nabla_J x_{i,j-1,k}) - (\nabla_K x_{i,j,k} - \nabla_K x_{i,j-1,k}).$$

(4.4)

The first step of a primal-dual algorithm is to decide a term converted into a form of its convex conjugates using Legendre-Fenchel transform to formulate a saddle point problem. Because a $l_1$ norm penalty term is difficult to solve in a primal minimization problem due to its non-smoothness, we need to find a convex conjugate of TV term. Compared to eq.2.24, the convex conjugates of $f_2$ should be derived:

$$f^*_2(q) = \max_x (q, \nabla x) - \lambda \|\nabla x\|_1$$

$$= \max_x \sum_i q_i (\nabla x)_i - \lambda \sum_i |(\nabla x)_i|$$

$$= \delta_{\text{Box}(\lambda)}(q),$$

(4.5)

where $\delta_{\text{Box}(\lambda)}(|q|)$, a indicator function, denotes that if magnitudes of all voxels in $q$ is less than $\lambda$, the value of a function becomes 0 otherwise $\infty$, that is,

$$\delta_{\text{Box}(\lambda)}(|q|) = \begin{cases} 0, & \forall (q)_i \leq \lambda \\ \infty, & \text{otherwise} \end{cases}.$$  

(4.6)
Then, a saddle-point problem is set up:

$$\min_{x} \max_{q} \|Ax - b\|^2 + \langle q, \nabla x \rangle - \delta_{\Box}(|q|).$$ \hspace{1cm} (4.7)

The derivation of the proximal mapping of $f_1$ and $f_2$ is the final step of a primal-dual algorithm. Using eq.2.25, a solution of the dual maximization problem is obtained as the following form:

$$q_{n+1} = \text{prox}_\sigma[f_2^*](\tilde{q}_n)$$

$$= \arg\min_q \left\{ \frac{1}{2\tau}||q - \tilde{q}_n||^2_2 + \delta_{\Box}(|q|) \right\},$$

where $\tilde{q}_n = q_n + \sigma \nabla x_n$. To minimize a cost function in eq.4.8, the magnitude of all voxels in $q$ should be less than $\lambda$ for setting the value of an indicator function to 0, not $\infty$. Under this constraint, $q$ is decided by size comparison of $\lambda$ and the absolute value of $\tilde{q}_n$ to minimize the first term. Then, the proximal mapping is computed:

$$(q_n)_i = \frac{\lambda(\tilde{q}_n)_i}{\max(\lambda, |(\tilde{q}_n)_i|)}.$$ \hspace{1cm} (4.9)

A primal minimization problem is easy to solve by solvable using eq.2.26 because $f_1$ is differentiable:

$$x_{n+1} = \text{prox}_\tau[f_1](\bar{x}_n)$$

$$= \arg\min_x \left\{ \frac{1}{2\tau}||x - \bar{x}_n||^2_2 + \|Ax - b\|^2 \right\}$$

$$= \left( A^H A + \frac{1}{2\tau}I \right)^{-1} \left( A^H b + \frac{1}{2\tau} x_n \right),$$ \hspace{1cm} (4.10)

where $\bar{x}_n = x_n + \tau \text{div} q_{n+1}$. The matrix inversion can be done efficiently using a conjugate gradient method. The resulting primal-dual algorithm for TV norm regularized problem is summarized as following:

<table>
<thead>
<tr>
<th>Algorithm6. Primal-Dual Algorithm for TV norm</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> $K = |\nabla|_2$, $\sigma = \frac{1}{\kappa}$, $\tau = \frac{1}{\kappa}$, $\theta = 1$.</td>
</tr>
<tr>
<td><strong>Step 0. Initialize $x_1$ and $q_1$, $x_1 = x_1$.</strong></td>
</tr>
<tr>
<td><strong>Step n.</strong></td>
</tr>
<tr>
<td>$q_{n+1} = \frac{\lambda(q_n + \sigma \nabla x_n)}{\max(\lambda,</td>
</tr>
<tr>
<td>$x_{n+1} = \left( A^H A + \frac{1}{2\tau} I \right)^{-1} \left( A^H b + \frac{1}{2\tau} (x_n + \tau \text{div} q_{n+1}) \right)$</td>
</tr>
<tr>
<td>$\bar{x}<em>{n+1} = x</em>{n+1} + \theta(x_{n+1} - x_n).$ \hspace{1cm} (4.13)</td>
</tr>
</tbody>
</table>

In a primal-dual algorithm, most computationally expensive step is to calculate the proximal mapping for the primal variable $x$, which is solved by using conjugate gradient method.

The convex conjugate of TV-norm can be used in FISTA. The main advantage of FISTA is the rapid convergence rate with computational simplicity by introducing $y$ for an extra-gradient step. However, limited application is an evident weakness compared to a primal-dual algorithm and ADMM. In this study, we attempt to connect FISTA with a primal-dual algorithm which can be applied to a general case of TV norm optimization problems to overcome the weakness of FISTA.

Eq.4.3 is transformed by using eq.2.11

$$x_{n+1} = \text{prox}_{1/L}[f_2](x_n - \frac{1}{L} \nabla f_1(x_n))$$

$$= \arg\min_x \left\{ \frac{L}{2} \|x - (x_n - \frac{1}{L} \nabla f_1(x_n))\|^2_2 + f_2(x) \right\}$$

$$= \arg\min_x \left\{ \frac{L}{2} \|x - (x_n - \frac{1}{L} A^H (Ax_n - b))\|^2_2 + \lambda \|\nabla x\|_1 \right\},$$ \hspace{1cm} (4.14)
where $\eta$ in eq.2.11 is substituted for $\frac{1}{L}$ in the same way as FISTA. This proximal mapping is considered as a primal minimization problem in the perspective of a primal-dual algorithm. Because the bottleneck in eq.4.14 is a $l_1$ norm regularized term, we set the relationship in the primal problem:

$$f_1(x) = \frac{L}{2} \| x - \tilde{x}_n \|_2^2,$$

$$f_2(\nabla x) = \lambda \| \nabla x \|_1,$$

where $\tilde{x}_n = x_n - \frac{2}{L} A^H (Ax_n - b)$. Then, the convex conjugate of $f_2$ results in the form of eq.4.5. Using this dual function, we formulate a saddle-point problem of eq.4.14:

$$\min_{x} \max_{q} \frac{L}{2} \| x - \tilde{x}_n \|_2^2 + \langle q, \nabla x \rangle - \delta_{\text{box} }(\lambda) (|q|).$$

To achieve a primal-dual algorithm in FISTA, we perform the proximal mapping to solve a primal and a dual problem. The proximal mapping on $f_2^*$ is derived from eq.4.9 and the proximal mapping on $f_1$ comes from the following derivation:

$$x_{n+1} = \text{prox}_{\tau f_1}(\tilde{x}_n) = \arg \min_{x} \left\{ \frac{1}{2\tau} \| x - \tilde{x}_n \|_2^2 + \frac{L}{2} \| x - \tilde{x}_n \|_2^2 \right\} = \left( \frac{\tau}{1 + \tau L} \right) (L\tilde{x}_n + \frac{1}{\tau} \tilde{x}_n).$$

Compared to eq.4.18, the matrix inversion is not necessary in eq.4.18 so that FISTA does not need to perform a conjugate gradient method to find a primal variable. All the take together, FISTA combined with a primal-dual algorithm results in the following steps:

**Algorithm 7. FISTA with a primal-dual algorithm for TV norm**

**Input:** $L$: the Lipschitz constant of $f_1$.

**Step 0.** Initialize $x_1, y_1 = x_1, q_1, t_1 = 1, K = \| \nabla \|_2, \sigma = \frac{1}{K}, \tau = \frac{1}{K}, \theta = 1$.

**Step n.**

$$\tilde{y}_n = y_n - \frac{2}{L} A^H (Ay_n - b) \quad (4.19)$$

$$x_m = x_n \quad (4.20)$$

**Step m.**

$$q_{m+1} = \frac{\lambda(q_m + \sigma \nabla x_m)}{\max\{\lambda, |q_m + \sigma \nabla x_m|\}},$$

$$x_{n+1} = \left( \frac{\tau}{1 + \tau L} \right) (L\tilde{y}_n + \frac{1}{\tau} (x_m + \tau \text{div } q_{m+1})), \quad (4.22)$$

$$x_{n+1} = x_{m+1} + \theta(x_{m+1} - x_m), \quad (4.23)$$

$$x_{n+1} = \hat{x}_{m+1}, \quad (4.24)$$

$$t_{n+1} = \frac{1 + \sqrt{1 + 4t_n^2}}{2} \quad (4.25)$$

$$y_{n+1} = x_{n+1} + \left( \frac{t_{n-1}}{t_n} \right) (x_{n+1} - x_n) \quad (4.26)$$

Introducing a primal-dual algorithm into FISTA produces pros and cons. The merit is to make up for the shortcomings of each algorithm. FISTA becomes useful with more various regularized function and a primal-dual algorithm requires much less computational efforts in FISTA for a primal variable. In algorithm 5, a primal-dual uses a conjugate gradient method, however, it is required only one step to find $x_{m+1}$ in algorithm 6. Moreover, it is expected that there is a more improved convergence rate since both algorithms have an extra-gradient step respectively. The disadvantage of this combined algorithm is to demand more memory space for storing a dual variable.
It is easy to solve eq.4.3 using ADMM. We choose a variable to split in the first step of ADMM. For computation simplicity, it is appropriate to substitute $\nabla x$ for $z$. Then, eq.4.3 is rewritten as the form of a constrained problem:

$$\min_{x,z} \|Ax-b\|_2^2 + \lambda \|z\|_1 \quad \text{subject to } z = \nabla x.$$ 

(4.27)

For this equation, augmented Lagrangian is defined as:

$$L(x, z, w, \rho) = \|Ax-b\|_2^2 + \lambda \|z\|_1 + \frac{\rho}{2} \|\nabla x - z - w\|_2^2$$

(4.28)

Due to convexity of eq.4.28, we can use alternating minimization with respect to $x$ and $z$. Even though a minimization problem against $x$ seems to be simple since all terms in the problem are convex and smooth functions, computation is not simple by ill-conditioned $A$ and $\nabla$. The approximation of $x$ is obtained as the form of

$$x_{n+1} = \arg \min_x \|Ax-b\|_2^2 + \frac{\rho}{2} \|\nabla x - z_n - w_n\|_2^2$$

(4.29)

$$z_{n+1} = \left\{ \begin{array}{ll} (|\tilde{z}_n| - \frac{1}{\rho}) \times sign(|\tilde{z}_n|), & \text{if } |\tilde{z}_n| \geq \frac{1}{\rho} \\ 0, & \text{otherwise} \end{array} \right.$$ 

(4.30)

where $\tilde{z}_n = \nabla x_{n+1} - w_n$. With these results, ADMM can be summarized as following:

**Algorithm 8. Alternating Direction Method of Multipliers for TV norm**

\textbf{Step 0.} Initialize $x_1, z_1 = \nabla x_1, w_1$.

\textbf{Step n.}

$$x_{n+1} = (2A^H A + \rho - \text{div} \nabla)^{-1}(2A^H b + \rho - \text{div}(z_n + w_n)),$$

(4.31)

$$z_{n+1} = \left\{ \begin{array}{ll} (|\tilde{z}_n| - \frac{1}{\rho}) \times sign(|\tilde{z}_n|), & \text{if } |\tilde{z}_n| \geq \frac{1}{\rho} \\ 0, & \text{otherwise} \end{array} \right.$$ 

(4.32)

$$w_{n+1} = w_n - (\nabla x_{n+1} - z_{n+1})$$

(4.33)

In Algorithm7, the most computationally intense step is the updating $x$, which is solved by using a conjugate gradient method similar to a primal-dual algorithm. However, unlike in a primal-dual algorithm, $\nabla$ is associated with approximation of $x$ in ADMM so that ADMM may make the computation more complicated. In addition, although the number of variables required to store for ADMM is smaller than that of FISTA and a primal-dual algorithm, the comparison of memory usage between algorithms is not possible because in ADMM, a splitted variable and a vector of Lagrange multipliers are transformed into a new domain by $\nabla$. In other two algorithms, only dual variable is transformed into a new domain among stored variables. Therefore, memory usage of ADMM is dependent on the size of a transformed domain.

### 4.2 Demonstration

Applying three algorithms, we first, reconstruct a down-sampled 3D phantom image by a factor of 3. Fig.4.1 displays a down-sampled image and reconstructed images using various algorithms. All of the images in fig.4.1 are almost perfectly reconstructed. MSE of the reconstucted images are near or below
$10^{-4}$. ADMM shows remarkable performance among three algorithms in fig.4.2 and table 4.1. Even though the initialization of $x$ is a zero matrix for all algorithms, ADMM searched the best approximation in the first step. In graphical analysis, the slope of a MSE curve for ADMM is similar with that of FISTA. Nevertheless, a MSE curve converges faster in ADMM than in FISTA because of first approximation. ADMM is numerically dominant in every criteria. In table 4.1, convergence time was measured when the change of MSE became below $10^{-5}$, and final MSE is the MSE in the last steps of iteration. Memory usage came from the difference between memory before debugging an algorithm and after debugging. ADMM has the fastest convergence rate and the closest approximation with the least memory usage.

The results of real data displayed different tendencies. The performance of all algorithms shows similar level where MSE of a reconstructed image is much lower than that of a down-sampled image as shown in fig.4.3 and 4.5. However, the quality of the image is distinguished from a full-sampled image in fig.3.1. Due to the similar quality, we displayed one maximum intensity projection of a sagittal plane where much more blood vessels are displayed. The optimization algorithm reconstructed lots of blood vessels which are disappeared in fig.4.4 (c) by down-sampling. In the MRA case, FISTA shows the fastest rate of convergence in fig.4.5. There is no algorithm to establish superiority in all criteria unlike the case of phantom reconstruction. While FISTA spent the shortest time to converge, ADMM found the most precise solution. A primal-dual algorithm used the smallest space to implement.

![Image](image.png)

Figure 4.1: (a) 64-th frame of a down-sampled 3D Shepp-Logan phantom image by a factor of 3, and a reconstructed image using (b) a primal-dual algorithm, (c) FISTA and (d) ADMM
Figure 4.2: mse curve versus computational time for a primal-dual algorithm (red), FISTA (blue) and ADMM (green) while reconstructing a down-sampled 3D Shepp-Logan image

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Convergence time</th>
<th>Final MSE</th>
<th>Memory usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primal-dual</td>
<td>1502s</td>
<td>$2.692 \times 10^{-4}$</td>
<td>34.48MB</td>
</tr>
<tr>
<td>FISTA</td>
<td>646.4s</td>
<td>$2.836 \times 10^{-4}$</td>
<td>34.09MB</td>
</tr>
<tr>
<td>ADMM</td>
<td>462.8s</td>
<td>$8.964 \times 10^{-7}$</td>
<td>34.48MB</td>
</tr>
</tbody>
</table>

Table 4.1: Performance comparison of a primal-dual algorithm, FISTA and ADMM for reconstructing a down-sampled 3D Shepp-Logan image
Figure 4.3: (a) 64-th frame of a down-sampled MRA image by a factor of 3, and a reconstructed image using (b) a primal-dual algorithm, (c) FISTA and (d) ADMM
Figure 4.4: Maximum intensity projection images of (a) full-sampling data, (b) down-sampled data and reconstruction using (c) a primal-dual algorithm, (d) FISTA and (e) ADMM
Figure 4.5: mse curve versus computational time for a primal-dual algorithm (red), FISTA (blue) and ADMM (green) while reconstructing a MRA image.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Convergence time</th>
<th>Final MSE</th>
<th>Memory usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primal-dual</td>
<td>8814.3s</td>
<td>0.0233</td>
<td>115.3MB</td>
</tr>
<tr>
<td>FISTA</td>
<td>1975.3s</td>
<td>0.0235</td>
<td>3119MB</td>
</tr>
<tr>
<td>ADMM</td>
<td>5394.5s</td>
<td>0.0226</td>
<td>119.5MB</td>
</tr>
</tbody>
</table>

Table 4.2: Performance comparison of a primal-dual algorithm, FISTA and ADMM for reconstructing a down-sampled MRA image.
Chapter 5. Wavelet Transform

5.1 Algorithm Description

The wavelet transform is widely used for a sparse representation in the image processing field [20], [21]. If few coefficients cover most information in an image, processing becomes faster and simpler in sparse representation. In our application, the wavelet transform, $\psi$ is attractive to adapt to $D$ for a sparse representation and the inverse wavelet transform, wavelet reconstruction $\psi^H$, plays a role of $D^H$.

Then, the wavelet regularized problem is formulated as following

\[
\min_x \|Ax-b\|_2 + \lambda \|\psi x\|_1.
\]

(5.1)

The form of eq.5.1 is quite similar to TV application so that algorithm is also derived in the same way. When a regularized term is set as $f_2$, the convex conjugate of $f_2$ is identical to eq.4.5. A saddle point problem of a primal minimization problem eq.5.1 is:

\[
\min_x \max_q \|Ax-b\|_2^2 + \langle q, \psi x \rangle - \delta_{Box}(\lambda)(|q|).
\]

(5.2)

For this saddle point problem, a primal-dual algorithm is developed as following:

\[
\text{Algorithm 9. Primal-Dual Algorithm for wavelet norm}
\]

\begin{itemize}
  \item Input : $K = \|\psi\|_2, \sigma = \frac{1}{K}, \tau = \frac{1}{K}, \theta = 1$.
  \item Step 0. Initialize $x_1$ and $q_1$, \(\bar{x}_1 = x_1\).
  \item Step n.
    \begin{align*}
    q_{n+1} &= \frac{\lambda(q_n + \sigma \psi x_n)}{\max(\lambda, q_n + \sigma \psi x_n)} \\
    x_{n+1} &= \left( A^H A + \frac{1}{\tau L} I \right)^{-1} \left( A^H b + \frac{1}{\tau} (x_n - \tau \psi^H q_{n+1}) \right) \\
    \bar{x}_{n+1} &= x_{n+1} + \theta (x_{n+1} - x_n).
    \end{align*}
\end{itemize}

A primal-dual algorithm is introduced into FISTA to solve a regularized term as shown in TV application. In wavelet application, a primal-dual algorithm is also required to develop FISTA. Algorithm 9 is almost same with Algorithm 6 except for a transformation operator. Hence, FISTA is as well designed as following:

\[
\text{Algorithm 10. FISTA with a primal-dual algorithm for wavelet norm}
\]

\begin{itemize}
  \item Input : $L$: the Lipschitz constant of $f_1$.
  \item Step 0. Initialize \(x_1, y_1 = x_1, q_1, t_1 = 1, K = \|\psi\|_2, \sigma = \frac{1}{K}, \tau = \frac{1}{K}, \theta = 1\).
  \item Step n.
    \begin{align*}
    \hat{y}_n &= y_n - \frac{\tau}{2} A^H (A y_n - b) \quad (5.6) \\
    x_m &= x_n (5.7) \\
    \bar{x}_m &= x_{m+1} + \theta (x_{m+1} - x_m). (5.10)
    \end{align*}
\end{itemize}

- 23 -
For ADMM, $\psi x$ is substituted into a splitted variable $z$. Then, ADMM for wavelet norm is also designed to totally same steps with those of TV except for $\psi$ as following:

Algorithm 11. Alternating Direction Method of Multipliers for wavelet norm

**Step 0.** Initialize $x_1, z_1 = \psi x_1, w_1$.

**Step n.**

$$x_{n+1} = (2A^HA + \rho + \psi H\psi)^{-1}(2A^Hb + \rho + \psi H(z_n + w_n)),$$

$$z_{n+1} = \begin{cases} (|\hat{z}_n| - \frac{\lambda}{\rho}) \times \text{sign}(|\hat{z}_n|), & \text{if } |\hat{z}_n| \geq \frac{\lambda}{\rho} \\ 0, & \text{otherwise} \end{cases}$$

$$w_{n+1} = w_n - (\psi x_{n+1} - z_{n+1})$$

5.2 Demonstration

Based on this algorithm development, we implemented 3D wavelet decomposition with db5 filter and level 3. We examined algorithms in reconstruction of a down-sampled 3D phantom image by a factor of 3. Fig.5.1 displays a down-sampled image and reconstructed images using three algorithms. All of the images in fig.5.1 are almost perfectly reconstructed like TV application. However, the performance of each algorithm is completely different with TV case. As shown in fig.5.2 and table 5.1, a primal-dual algorithm has the fastest rate of convergence in wavelet application. Identical to TV application, ADMM performed the smallest MSE and used the least memory. In this case, a primal-dual algorithm searched better approximation in the first step than FISTA. This became the critical factor of great performance because in graphical analysis, the slope of a MSE curve for a primal-dual algorithm is similar with that of FISTA. Like the manner of table 4.1, in table 5.1, convergence time was measured when the change of MSE became below $10^{-5}$, and final MSE is the MSE in the last steps of iteration. Memory usage came from the difference between memory before debugging an algorithm and after debugging. ADMM has the fastest convergence rate and the closest approximation with the least memory usage.

The wavelet application of real data also displayed different results with TV application. The performance of all algorithms are similar level where MSE of a reconstructed image is much lower than MSE of a down-sampled image as shown in fig.5.3 and 5.5 like TV application. However, The quality of the image is worse than that of reconstructed images using TV. Final MSE in wavelet application is generally more than MSE in TV application. The convergence rate of FISTA is noticeably slow compared to other two algorithms. ADMM showed outstanding performance including the shortest convergence time, the minimum final MSE and the least memory usage. We displayed only maximum intensity projection of a sagittal plane using a primal-dual algorithm. The optimization algorithm reconstructed some blood vessels which are disappeared in fig.5.4 (c) by down-sampling.
Figure 5.1: (a) 64-th frame of a down-sampled 3D Shepp-Logan phantom image by a factor of 3, and a reconstructed image using (b) a primal-dual algorithm, (c) FISTA and (d) ADMM

Table 5.1: Performance comparison of a primal-dual algorithm, FISTA and ADMM for reconstructing a down-sampled 3D Shepp-Logan image

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Convergence time</th>
<th>Final MSE</th>
<th>Memory usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primal-dual</td>
<td>810.6s</td>
<td>0.0184</td>
<td>31MB</td>
</tr>
<tr>
<td>FISTA</td>
<td>752.6s</td>
<td>0.0187</td>
<td>33MB</td>
</tr>
<tr>
<td>ADMM</td>
<td>906.58s</td>
<td>0.0077</td>
<td>27.1MB</td>
</tr>
</tbody>
</table>
Figure 5.2: mse curve versus computational time for a primal-dual algorithm (red), FISTA (blue) and ADMM (green) while reconstructing a down-sampled 3D Shepp-Logan image.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Convergence time</th>
<th>Final MSE</th>
<th>Memory usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primal-dual</td>
<td>946.46s</td>
<td>0.0304</td>
<td>95.85MB</td>
</tr>
<tr>
<td>FISTA</td>
<td>3885.5s</td>
<td>0.0306</td>
<td>115.7MB</td>
</tr>
<tr>
<td>ADMM</td>
<td>589.4s</td>
<td>0.0286</td>
<td>122.4MB</td>
</tr>
</tbody>
</table>

Table 5.2: Performance comparison of a primal-dual algorithm, FISTA and ADMM for reconstructing a MRA image.
Figure 5.3: (a) 64-th frame of a down-sampled MRA image by a factor of 3, and a reconstructed image using (b) a primal-dual algorithm, (c) FISTA and (d) ADMM
Figure 5.4: Maximum intensity projection images of (a) full-sampling data, (b) down-sampled data and reconstruction using (c) a primal-dual algorithm, (d) FISTA and (e) ADMM
Figure 5.5: mse curve versus computational time for a primal-dual algorithm (red), FISTA (blue) and ADMM (green) while reconstructing a MRA image
Chapter 6. Patch-based Low Rank Approximation

6.1 Algorithm Description

Biological structure is continuous inside boundary such as membranes. Especially, same type of tissue cells are in a cluster in a tissue to play its own function. Hence, a small region of biological structure has a similar shape with neighborhood. An application using this characteristic is patch-based low rank approximation. Patch-based low rank approximation is based on finding the most similar patches in the neighbor frames and performing low rank approximation to improve the image quality. The scheme of an application is introduced below.

First, we construct a patch and calculate the difference within other patches in a decided window. We, then, extract patches which have the least difference. A matrix is constituted to the most similar patches along multiple slice trajectory. A similar patch becomes vectorized and located in a vector of the matrix. If we set up a patch size $3 \times 3$ and choose 4 similar patches, the size of a matrix become $9 \times 4$. After constructing a matrix, we perform low rank approximation of the matrix to extract the important features as shown in fig.6.2.

For this application, eq.2.6 should be modified as following:

$$\min_{X} \|AX - B\|_F^2 + \lambda \sum_p \|D_pX\|_*, \quad (6.1)$$

where $D_p$ constructs a matrix by finding similar patches with a patch of a $p$-th voxel. Then, $D_p^T$ returns a vector of a matrix into the original position existing a patch.

Eq.6.1 is much more complicated than $l_1$ norm regularized problems. The reason is that in a nuclear norm problem, the fidelity term is calculated by the original matrix form, whereas nuclear norm in the regularized term is concerned with only singular values. Thus, we should change an optimization problem with respect to a matrix like the fidelity term of eq.6.1 into a problem against its singular values.

Here is an example problem:

$$\min_{S} \|S - T\|_F^2, \quad (6.2)$$
Figure 6.2: Low rank approximation

where the SVD of \( S \) and \( T \) is considered as \( S = U_S \Sigma_S V_S \) and \( T = U_T \Sigma_T V \) respectively. Since \( U_S, V_S, U_T \) and \( V_T \) are unitary matrices, eq.6.2 can be reformulated as:

\[
\min_{\Sigma_S} \|U_S^H \Sigma_T V_T - \Sigma_T\|_F^2
\]

\[
= \min_{\Sigma_S} \|\Sigma_T - U_S^H U_S \Sigma_S V_T V_T^H \|_F^2.
\]

The second term in the Frobenius norm should be maximized for minimizing the Frobenius norm. All \( U \) and \( V \) are unitary so that the following Cauchy-Schwarz inequalities are satisfied:

\[
|\langle u_T, u_S \rangle|^2 \leq \langle u_T, u_T \rangle \cdot \langle u_S, u_S \rangle = 1,
\]

\[
|\langle v_S, v_T \rangle|^2 \leq \langle v_S, v_S \rangle \cdot \langle v_T, v_T \rangle = 1,
\]

where \( u \) and \( v \) are vectorized forms of \( U \) and \( V \). It means that eq.6.3 is minimized when \( U_S^H U_S = V_S^H V_T = I \). These relationships, eventually, makes that eq.6.2 becomes identical to the following problem involving only the singular values of \( S \):

\[
\min_{\Sigma_S} \|\Sigma_S - \Sigma_T\|_F^2.
\]

The derivation of a transformed problem with respect to its singular values is quite helpful to obtain the convex conjugate of nuclear norm. To find a solution to eq.6.1 using a primal-dual algorithm, we need to derive \( f_p^* \) after substituting \( D_p x \) into \( S_p \) for simplicity. The convex conjugate is produced by the following background:

\[
f_p^*(Q_p) = \max_{S_p} \langle Q_p, S_p \rangle - f_2(S_p)
\]

\[
= \max_{S_p} \frac{1}{2} \|S_p\|_F^2 + \frac{1}{2} \|Q_p\|_F^2 - \frac{1}{2} \|S_p - Q_p\|_F^2 - \lambda \|S_p\|_S
\]

\[
= \max_{\Sigma_s} \frac{1}{2} \|\Sigma_s\|_F^2 + \frac{1}{2} \|Q_p\|_F^2 - \frac{1}{2} \|\Sigma_s - \Sigma Q_p\|_F^2 - \lambda \text{tr}(\Sigma_s^2)
\]

\[
= \max_{\Sigma_s} \frac{1}{2} \text{tr}(\Sigma_s^2) + \frac{1}{2} \text{tr}(\Sigma Q_p^2) - \frac{1}{2} \text{tr}((\Sigma_s - \Sigma Q_p)^2) - \lambda \text{tr}(\Sigma_s^2)
\]

\[
= \max_{\Sigma_s} \text{tr}((\Sigma Q_p - \lambda I)\Sigma_s)
\]

\[
= \delta_{\text{Box}(\lambda)}(\|\Sigma Q_p\|_s) = \begin{cases} 0, & \forall (\Sigma Q_p)_i \leq \lambda \\ \infty, & \text{otherwise} \end{cases}
\]

- 31 -
where 'tr' is trace of a matrix and \( \| \cdot \|_F^2 = \text{tr}(\cdot^2) \). Then, a saddle point problem of eq.6.1 is defined as:

\[
\min_{\mathbf{X}} \max_{\mathbf{Q}_p} \| \mathbf{A} \mathbf{X} - \mathbf{B} \|_F^2 + \sum_p \left( \| \mathbf{Q}_p \mathbf{D}_p \mathbf{X} \|_F^2 - \delta_{\text{Box}(\lambda)}(\| \Sigma \mathbf{Q}_p \|) \right). \tag{6.8}
\]

The final step for using a primal-dual algorithm is the proximal mapping on \( f_1 \) and \( f_2^* \). The fidelity term of eq.6.1 is exactly same with that of eq.4.3 so that the proximal mapping is also identical to eq.4.18. The proximal mapping on \( f_2^* \) becomes the following thresholding:

\[
\mathbf{Q}_{p,n+1} = \text{prox}_{\sigma f_2^*}(\tilde{\mathbf{Q}}_{p,n}) = \arg \min_{\mathbf{Q}_p} \left\{ \frac{\| \mathbf{Q}_p - \tilde{\mathbf{Q}}_{p,n} \|_F^2}{2\sigma} + \delta_{\text{Box}(\lambda)}(\| \Sigma \tilde{\mathbf{Q}}_{p,n} \|) \right\}
\]

\[
= \arg \min_{\Sigma_{\mathbf{Q}_p}} \left\{ \frac{\| \Sigma_{\mathbf{Q}_p} - \Sigma_{\tilde{\mathbf{Q}}_{p,n}} \|_F^2}{2\sigma} + \delta_{\text{Box}(\lambda)}(\| \Sigma_{\mathbf{Q}_p} \|) \right\}
\]

\[
= \mathbf{U}_{\mathbf{Q}_{p,n}} \Sigma_{\mathbf{Q}_{p,n}} \left( \frac{\Sigma_{\tilde{\mathbf{Q}}_{p,n}}}{\max(1, \| \Sigma_{\tilde{\mathbf{Q}}_{p,n}} \|)} \right) \mathbf{V}_{\mathbf{Q}_{p,n}}^H,
\tag{6.9}
\]

where \( \tilde{\mathbf{Q}}_{p,n} = \mathbf{Q}_{p,n} + \sigma \mathbf{D}_p \mathbf{X}_n \) and the SVD of \( \tilde{\mathbf{Q}}_{p,n} \) is \( \mathbf{U}_{\mathbf{Q}_{p,n}} \Sigma_{\mathbf{Q}_{p,n}} \mathbf{V}_{\mathbf{Q}_{p,n}}^H \). The reason why the approximation of a dual variable in a nuclear-norm problem is similar with that in a \( l_1 \) norm problem is that in a \( l_1 \) norm problem is that nuclear norm is also \( l_1 \) norm of its singular values. Therefore, the resulting algorithm is summarized as following:

**Algorithm 9. Primal-Dual Algorithm for patch-based low rank approximation**

**Input:** \( K = \| \mathbf{D} \|_2, \sigma = \frac{1}{K}, \tau = \frac{1}{K}, \theta = 1 \).

**Step 0.** Initialize \( \mathbf{X}_1 \) and \( \mathbf{Q}_{p,1} \), \( \mathbf{X}_1 = \mathbf{X}_0 \).

**Step n.**

**Step p.**

\[ \mathbf{D}_p \mathbf{X}_n \leftarrow \text{Construct a matrix to find similar patches with a patch of a } p \text{-th voxel} \]

\[ \mathbf{Q}_{p,n} = \mathbf{Q}_{p,n} + \sigma \mathbf{D}_p \mathbf{X}_n \]

\[ \mathbf{Q}_{p,n+1} = \mathbf{U}_{\mathbf{Q}_{p,n}} \Sigma_{\mathbf{Q}_{p,n}} \left( \frac{\Sigma_{\tilde{\mathbf{Q}}_{p,n}}}{\max(1, \| \Sigma_{\tilde{\mathbf{Q}}_{p,n}} \|)} \right) \mathbf{V}_{\mathbf{Q}_{p,n}}^H \tag{6.10} \]

\[ \mathbf{D}_p^H \mathbf{Q}_{p,n+1} \leftarrow \text{Return } \mathbf{Q}_{p,n+1} \text{ into an original position} \]

\[ \mathbf{X}_n = \mathbf{X}_n - \tau \sigma \sum_p \mathbf{D}_p^H \mathbf{Q}_{p,n+1} \]

\[ \mathbf{X}_{n+1} = (\mathbf{A}^H \mathbf{A} + \frac{1}{2\tau} \mathbf{I})^{-1} (\mathbf{A}^H \mathbf{B} + \frac{1}{2\tau} \mathbf{X}_n) \]

\[ \mathbf{X}_{n+1} = \mathbf{X}_{n+1} + \theta(\mathbf{X}_{n+1} - \mathbf{X}_n). \tag{6.12} \]

Because the convex conjugate of a nuclear norm is obtained, FISTA combined with a primal-dual algorithm can be used. Computing the SVD pays computational cost so that the fast algorithm is more required than in a \( l_1 \) norm problem. Theoretically, FISTA has faster convergence rate than a primal-dual algorithm because it performs an extra-gradient step twice. Quadratic approximation of eq.6.1 for using FISTA is

\[
\mathbf{X}_{n+1} = \arg \min_{\mathbf{X}} \left\{ \frac{L}{2} \| \mathbf{X} - (\mathbf{X}_n - \frac{2}{L} \mathbf{A}^H (\mathbf{A} \mathbf{X}_n - \mathbf{B}) \|_2^2 + \lambda \sum_p \| \mathbf{D}_p \mathbf{X} \|_* \right\}. \tag{6.13}
\]

Then, FISTA is performed similar with the case of \( l_1 \) norm:
Algorithm 10. FISTA with a primal-dual algorithm for patch-based low rank approximation

**Input**: $L$: the Lipschitz constant of $f_1$.

**Step 0.** Initialize $X_1, Y_1 = X_1, Q_1, t_1 = 1, K = \|D_p\|_2, \sigma = \frac{1}{\pi}, \tau = \frac{1}{\pi}, \theta = 1$.

**Step n.**

\[
\begin{align*}
\tilde{Y}_n &= Y_n - \frac{1}{L}A^H(AY_n - B) \quad (6.14) \\
X_m &= X_n(6.15)
\end{align*}
\]

**Step m.**

**Step p.**

\[
\begin{align*}
D_pX_m &= Q_{p,m} = Q_{p,m} + \sigma D_pX_m \\
Q_{p,m+1} &= U_{\tilde{Q}_{p,m}} \cdot \left(\frac{\lambda \Sigma_{p,m}}{\max(\lambda \Sigma_{p,m})}\right) \cdot V_{\tilde{Q}_{p,m}}^H \quad (6.16) \\
X_{m+1} &= (\frac{1}{\tau + \tau}) \left(\tilde{L} \tilde{Y}_n + \frac{1}{\tau}(X_m - \tau \sum_p D_p^H Q_{p,m+1})\right) \quad (6.17) \\
\hat{X}_{m+1} &= X_{m+1} + \theta(X_{m+1} - X_m)(6.18)
\end{align*}
\]

\[
X_{n+1} = \hat{X}_{m+1}(6.19)
\]

\[
t_{n+1} = \frac{1 + \sqrt{1 + 4t_n^2}}{2} \quad (6.20)
\]

\[
Y_{n+1} = X_{n+1} + \left(\frac{t_{n+1}}{t_{n+1}}\right)(X_{n+1} - X_n)(6.21)
\]

For the perspective of ADMM, the approach of applying the algorithm is simple. We choose $D_pX$ to split a variable and formulate the following constrained problem:

\[
\min_{X,Z} \left\|AX - B\right\|_F^2 + \lambda \sum_p \|Z_p\|_*, \quad \text{subject to} \quad Z_p = D_pX. \quad (6.22)
\]

Like FISTA and a primal-dual algorithm, it is identical to the updating step of $X$ and $W$. A conjugate gradient method is required for updating $X$ and a simple calculation is performed for updating a vector of Lagrangian multipliers $W$. We need to check just the updating step of a splitted variable $Z_p$:

\[
Z_{p,n+1} = \arg\min_{Z_p} \lambda \sum_p \|Z_p\|_* + \frac{\rho}{2}\|Z_p - (D_pX_{n+1} - W_{p,n})\|_F^2
\]

\[
= \arg\min_{\Sigma_{Z_p}} \lambda \operatorname{tr}(\Sigma_{Z_p}) + \frac{\rho}{2}\|\Sigma_{Z_p} - \Sigma_{Z_{p,n}}\|_F^2
\]

\[
= \Sigma_{Z_{p,n}} - \frac{\lambda}{\rho}I \quad (6.23)
\]

where $\tilde{Z}_{p,n} = D_pX_{n+1} - W_{p,n}$. Using this derivation, ADMM for nuclear norm is summarized as following:

Algorithm 11. Alternating Direction Method of Multipliers for nuclear norm

**Step 0.** Initialize $X_1, Z_{p,1} = D_pX_1, W_{p,1}$.

**Step n.**

\[
\begin{align*}
X_{n+1} &= (2A^H + \rho D_p^H D_p)^{-1}(2A^H B + \rho D_p^H (Z_{p,n} + W_{p,n})), \quad (6.24) \\
Z_{p,n+1} &= \Sigma_{Z_{p,n}} - \frac{\rho}{\lambda}I \quad (6.25) \\
W_{p,n+1} &= W_{p,n} - (D_pX_{n+1} - Z_{p,n+1}) \quad (6.26)
\end{align*}
\]

6.2 Demonstration

We used MEX function in MATLAB to utilize c++ language unlike TV and wavelet application because it spent a lot of time to find similar patches for all voxels using MATLAB language. Consequently,
it is meaningless to compare computational time with other two algorithms. In this application, we need to store a matrix for all voxels which requires huge amount of memory. FISTA and ADMM are required too much memory space to implement the application based on patch-based low rank approximation. For this reason, it was impossible to implement ADMM and FISTA for patch-based low rank approximation by a used computer due to out of memory error. The reconstructed images of a 3D phantom and MRA are displayed below.

Figure 6.3: (a) 64-th frame of an original 3D Shepp-Logan phantom image by a factor of 3, (b) down-dampled image and (c) a reconstructed image using a primal-dual algorithm

Figure 6.4: (a) 150-th frame of an original MRA phantom image by a factor of 3, (b) down-dampled image and (c) a reconstructed image using a primal-dual algorithm
Figure 6.5: Maximum intensity projection images of (a) full-sampling data, (b) down-sampled data and (c) reconstruction using a primal-dual algorithm.
Chapter 7. Discussion

The implementation using a primal-dual algorithm, FISTA and ADMM brought good results. In phantom reconstruction case, all optimization problems based on total variation, wavelet transform and patch-based low rank approximation found almost true solution. Aliasing artifact were disappeared and the detailed shape like blood vessels became completely distinguishable. As a result, MSE converges to nearly 0. Improved image quality were also shown in the case of real data. Even though the quality of a reconstructed image did not reach to the level of a true image, it is clinically and academically more meaningful than a down-sampled image. First, most of aliasing patterns were removed. Second, compared to down sampled data, many blood vessels had faded in maximum intensity projection which is important thing in the perspective of TOF MRA. Because MRA research groups are still actively dealing with the reduction of scan-time, the result of this study is remarkable in MRA field.

The quality of resulting images depends on the regularization term. The widely known role of TV is smoothing an image while preserving edge. This role produced not only smoothness of tissues in the brain image but also restoration of missing blood vessels. In wavelet regularization, a filter and the level of wavelet decomposition have effects on image reconstruction. It seems to be important to choose suitable conditions depending on raw data. Patch-based low rank approximation deals with plural voxels to reconstuct one voxel. It was good for reconstruct overall shape of a voxel image because of information from similar patches. However, there was limitation to restore an image in detail.

In the perspective of an algorithm, ADMM showed the best performance among three algorithms, particularly in phantom image reconstruction. ADMM became slower and took more memory space in the case of real data reconstruction. The main reason of this change according to data is the transformation operator $D$. A primal-dual algorithm and FISTA calls for $D$ and $D^H$ once, respectively in one step, but ADMM uses them in conjugate gradient steps. That is, $D$ and $D^H$ are utilized much more times in ADMM than in others. Since computational effort of $D$ is proportional to the size of an image, ADMM performed in a better way in the case of a phantom of which size is $128 \times 128 \times 128$ compared to $512 \times 192 \times 76$ of MRA data. Memory usage also becomes the disadvantage of ADMM when bigger size of an image is treated. Even in patch-based low rank approximation, ADMM can not be applied because of memory problem. FISTA combined with a primal-dual algorithm is one of the noticeable points in this study. FISTA showed faster convergence rate than a primal-dual algorithm in some cases. In the previous studies, FISTA is usually the slowest algorithm among them. Therefore, the proper combination of algorithms brings on better performance depending on data and application.

The most difficult process during experiments was $\lambda$ optimization. We tried to find an optimal regularization parameter by changing $\lambda$ gradually. Each algorithm is very sensitive to $\lambda$ and the optimal value is different for different data and applications. We had to find $\lambda$ empirically. All experiments were implemented with the most optimal $\lambda$ we tried.
<table>
<thead>
<tr>
<th>Variables to necessarily store</th>
<th>FISTA</th>
<th>Primal-dual</th>
<th>ADMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{n+1}, x_n, y_n \in \mathbb{R}^n$</td>
<td>$q_n \in \mathbb{R}^m$</td>
<td>$x_{n+1}, x_n, y_n \in \mathbb{R}^n$</td>
<td>$q_n \in \mathbb{R}^m$</td>
</tr>
<tr>
<td>Call for operators in one step</td>
<td>$A$</td>
<td>1</td>
<td>2 x inner steps</td>
</tr>
<tr>
<td></td>
<td>$A^H$</td>
<td>1</td>
<td>1 x inner steps</td>
</tr>
<tr>
<td></td>
<td>$D$</td>
<td>1 x inner steps</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>$D^H$</td>
<td>1 x inner steps</td>
<td>1</td>
</tr>
<tr>
<td>TV for phantom</td>
<td>Time</td>
<td>Intermediate</td>
<td>slowest</td>
</tr>
<tr>
<td></td>
<td>MSE</td>
<td>biggest</td>
<td>intermediate</td>
</tr>
<tr>
<td></td>
<td>Memory</td>
<td>intermediate</td>
<td>most</td>
</tr>
<tr>
<td>TV for MRA</td>
<td>Time</td>
<td>fastest</td>
<td>slowest</td>
</tr>
<tr>
<td></td>
<td>MSE</td>
<td>biggest</td>
<td>intermediate</td>
</tr>
<tr>
<td></td>
<td>Memory</td>
<td>intermediate</td>
<td>least</td>
</tr>
<tr>
<td>Wavelet for phantom</td>
<td>Time</td>
<td>slowest</td>
<td>fastest</td>
</tr>
<tr>
<td></td>
<td>MSE</td>
<td>biggest</td>
<td>intermediate</td>
</tr>
<tr>
<td></td>
<td>Memory</td>
<td>most</td>
<td>intermediate</td>
</tr>
<tr>
<td>Wavelet for MRA</td>
<td>Time</td>
<td>slowest</td>
<td>fastest</td>
</tr>
<tr>
<td></td>
<td>MSE</td>
<td>biggest</td>
<td>intermediate</td>
</tr>
<tr>
<td></td>
<td>Memory</td>
<td>intermediate</td>
<td>least</td>
</tr>
</tbody>
</table>

Figure 7.1: Comparison of algorithm performance
Chapter 8. Conclusion

In this study, we compared recently developed three proximal algorithms such as FISTA, a primal-dual algorithm and ADMM, both theoretically and experimentally in various applications. First of all, we derived three algorithms with respect to a general case of $l_1$ norm and nuclear norm connected to a transform operator $D$. Then, total variation, wavelet transformation and patch-based low rank approximation were applied to $D$ for sparse representation. Sparse representation is one of the most important conditions of CS technique. Using this condition, all applications resulted in perfect reconstruction of a 3-folded down-sampled 3D Shepp-Logan phantom image. In the case of real data, image reconstruction using three algorithms improved the quality of an image rather than that of an image reconstructed by directly applying inverse Fourier transform. Even though ADMM showed the best performance among them, there were some limitations in the view of computational time and memory usage to apply ADMM as the dimension of an image became higher. We expect that the derivation and interpretation of algorithms according to application are helpful to choose an algorithm to apply in a problem of 3D-CS MR reconstruction.
References


Summary

Comparative studies on proximal algorithms for 3D MRI reconstruction

의 발전으로 인해 MRI의 환영 시간 감소 및 다운샘플링 된 데이터로부터의 고해상도 영상 복원이 가능해졌다. 최근에는 다양한 proxy 근접 알고리즘들이 개발되면서 더욱 빠른 속도와 높은 효율성으로 압축 센싱 이론을 적용할 수 있는 배경이 마련되고 있다. 근래 가장 각광 받고 있는 대표적인 proxy 알고리즘으로 FISTA, ADMM, primal-dual algorithm이 존재한다. 각각의 알고리즘은 어플리케이션에 따라 수렴 속도, 메모리 사용 등에 있어서 상이한 성능을 보인다. 본 연구에서는 3차원 MRI 정적 영상 데이터에 다양한 어플리케이션을 적용하였을 때 각 proxy 알고리즘들이 어떠한 성능을 나타내는지 비교하고, 장단점을 분석하여 어플리케이션 별 최적화된 proxy 알고리즘을 제시하는 것을 목적으로 하고 있다.